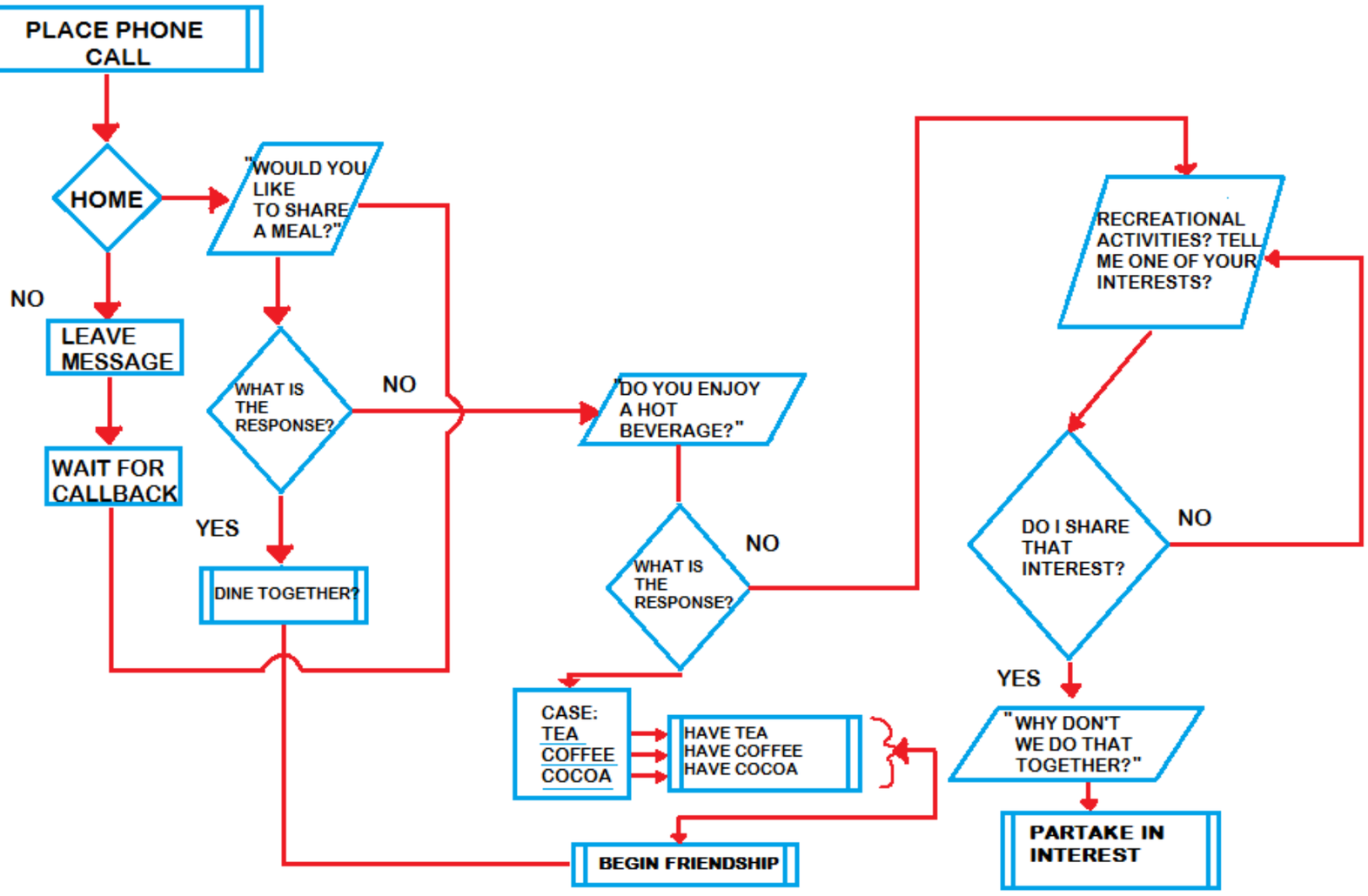
Wie sieht ein Algorithmus aus?

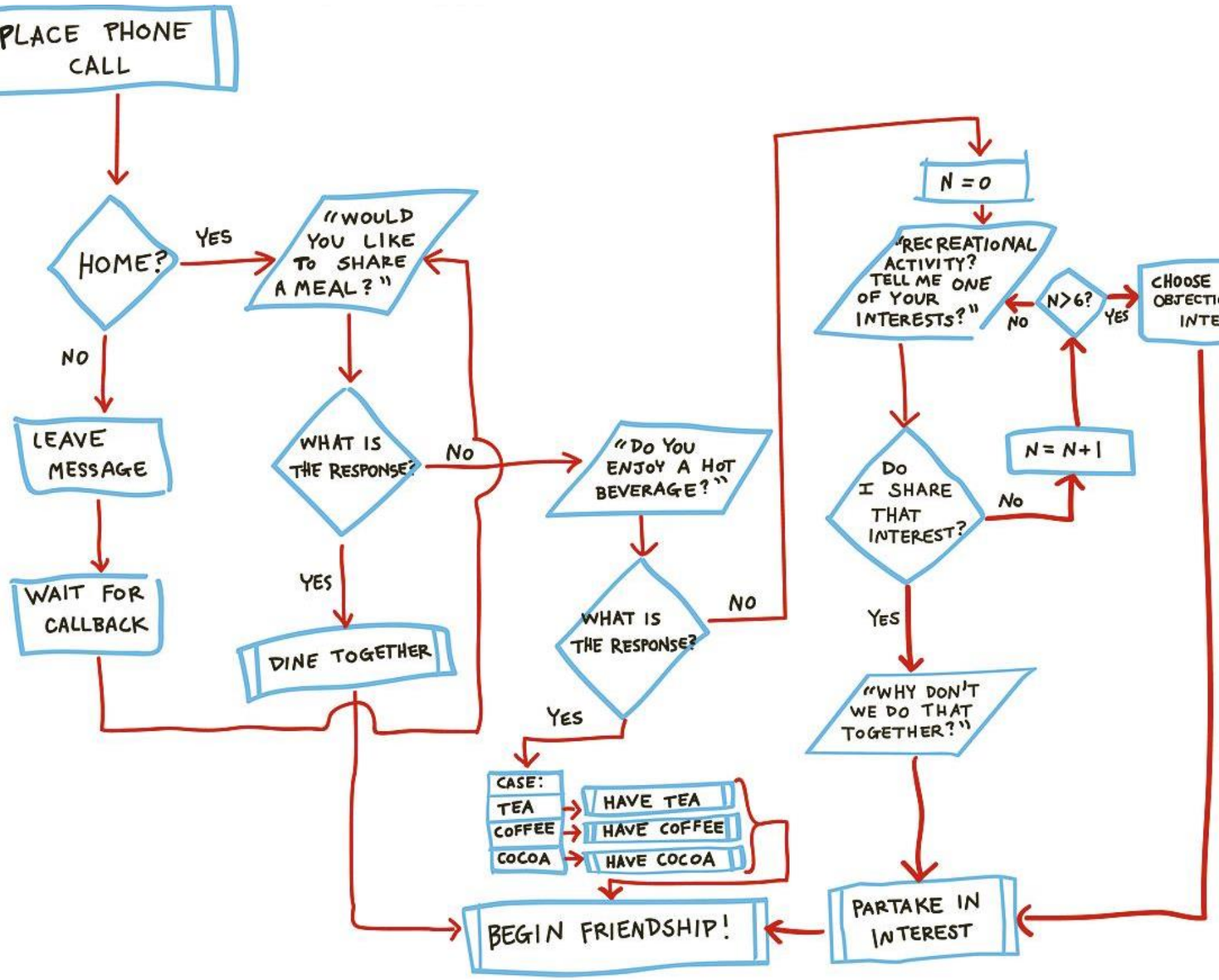
Der Freundschafts Algorithmus:

https://www.youtube.com/watch?v=I37XYZe8L-o

Freundschafts Algorithmus Original by Sheldon



Freundschafts-Algorithmus modifiziert by Wolowitz



Kontrollstrukturen in der Programmierung

In der Programmierung gibt es **3 Kontrollstrukturen**.

Mit diesen 3 Kontrollstrukturen kann in der Theorie jedes Programm das existiert geschrieben werden.

Sequenz

Verzweigung

Schleife

Sequenz

Alles, was wir bisher geschrieben haben.

Das Programm wird von oben nach unten **Zeile für Zeile** abgearbeitet.

https://scratch.mit.edu/

Verzweigung I

- Immer, wenn unser Programm eine **Entscheidung treffen** muss benötigen wir eine Verzweigung.

- Welche Entscheidung muss

  - ein Kaffeeautomat
  - ein Drucker
  - ein Auto

  treffen?

Verzweigung – Beispiel 5_1

5_1Verzweigung.py

```
1  import random
2
3  z1 = random.randint(1,10)
4  z2 = random.randint(1,10)
5
6  print("Die erste Zufallszahl ist " + str(z1) + "\nDie zweite Zufallszahl ist " + str(z2))
7
8  if (z1 > z2):
9      print("Die erste Zufallszahl war größer!")
10 else:
11     print("Die zweite Zufallszahl war größer!")
```

Shell

```
>>> %Run 5_1Verzweigung.py
Die erste Zufallszahl ist 5
Die zweite Zufallszahl ist 1
Die erste Zufallszahl war größer!
```

Das Programm funktioniert, jedoch ist trotzdem ein Problem vorhanden.
Kannst du das Problem erkennen?

Verzweigung – logische Operatoren

Die Bedingung im Kopf einer Verzweigung kann **True oder False** sein:

True => Ausführung des **if-Zweigs**

False => Ausführung des **else-Zweigs** (falls vorhanden)

Wir verwenden zum Vergleichen von 2 Werten folgende Operatoren:

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Quelle: https://www.w3schools.com/python/python_operators.asp